

Patent Claims

1. A method of loading a file from a distributed file system into a client comprising the steps of:

- providing a cache memory for temporarily storing loaded files;
- prior to each loading process, checking the cache memory to see whether a file to be loaded is present in the cache memory or not;
- loading the file from the cache memory, if the file is in the cache memory; and loading the file from the distributed file system otherwise;

wherein the step of checking the cache memory comprises computing a hash code from the file to be loaded; and checking the cache memory by means of the hash code, to see whether the file is contained in the cache memory.

2. A method as claimed in claim 1 wherein the distributed file system is a network with a plurality of servers.

3. A method as claimed in claim 2 wherein the network is the Internet or an intranet.

4. A method as claimed in claim 1 wherein the cache memory is provided in a proxy server interposed between the distributed file system and the client.

5. A method as claimed in claim 1 wherein the cache memory is provided in the client.

6. A method as claimed in claim 1 wherein the hash code is computed via the MD5 algorithm.

7. A method as claimed in claim 1 wherein the client sends a request to a server from which the file is to be loaded, wherein the server then computes and returns the hash code, and wherein subsequently the checking of the cache contents is done by means of the hash code.

8. A method as claimed in claim 1 wherein the hash code is defined in a hypertext link pointing at the file to be loaded.

9. A method as claimed in claim 1 wherein if the file to be loaded is not contained in the cache memory, a request is made to a search engine to retrieve an address at which the file can be found in the distributed file system.

10. A client for loading files from a distributed file system, particularly from the Internet, comprising

- a main memory for storing the files;
- an interface to the file system,
- a cache memory for temporarily storing loaded files; and

- a control unit for controlling the loading of the files and for controlling the cache memory, said control unit being adapted to check prior to each loading of a file whether the file to be loaded is present in the cache memory or not and, if the file is in the cache memory, to load it from the cache memory and otherwise to load it from the distributed file system,

wherein the control unit is adapted to perform the check as to whether the file to be loaded is present in the cache memory or not by means of a hash code computed from the file.

11. A proxy server comprising
 a cache memory for temporarily storing files to be loaded from a distributed file system into a client;
 a first interface to the client;
 a second interface to the distributed file system; and
 a control unit for controlling the loading of the files and for controlling the cache memory, said control unit being adapted to check prior to each loading of a file whether the file to be loaded is present in the cache memory or not and, if the file is in the cache memory, to send it from the cache memory via the first interface to the client and otherwise to load the file from the distributed file system via the second interface and send it to the client via the first interface;
 wherein the control unit is further adapted to perform the check as to whether the file to be loaded is present in the cache memory or not by means of a hash code computed from the file.

12. A server of a distributed file system, particularly a file server or Web server, which is adapted to make files available to clients via a network for loading, comprising an arithmetic unit for computing a hash code for each of the files.

13. Search engine for finding an address at which a file to be loaded is located in a distributed file system, comprising

- a receiver for receiving a query defining a hash code of the file to be loaded;
- a comparator for comparing the hash defined in the query with hash codes of files registered with the search engine and for retrieving the address of the file to be loaded if the comparison yields a file with the same hash code; and
- transmitter for transmitting back a query result containing the address of the file to be loaded if this is found and an error message otherwise.